Data Communication and Net-Centric Computing

COSC 1111/2061/1110

Lecture 6

Flow control, Error control

Lecture Overview

- During this lecture, we will understand
  - Flow control
  - Error Control
  - Look at Flow control that enables receiver to regulate the flow of data
  - Utilization of Flow control mechanisms

- Recommended reading
  - Chapter 7 (Stallings)

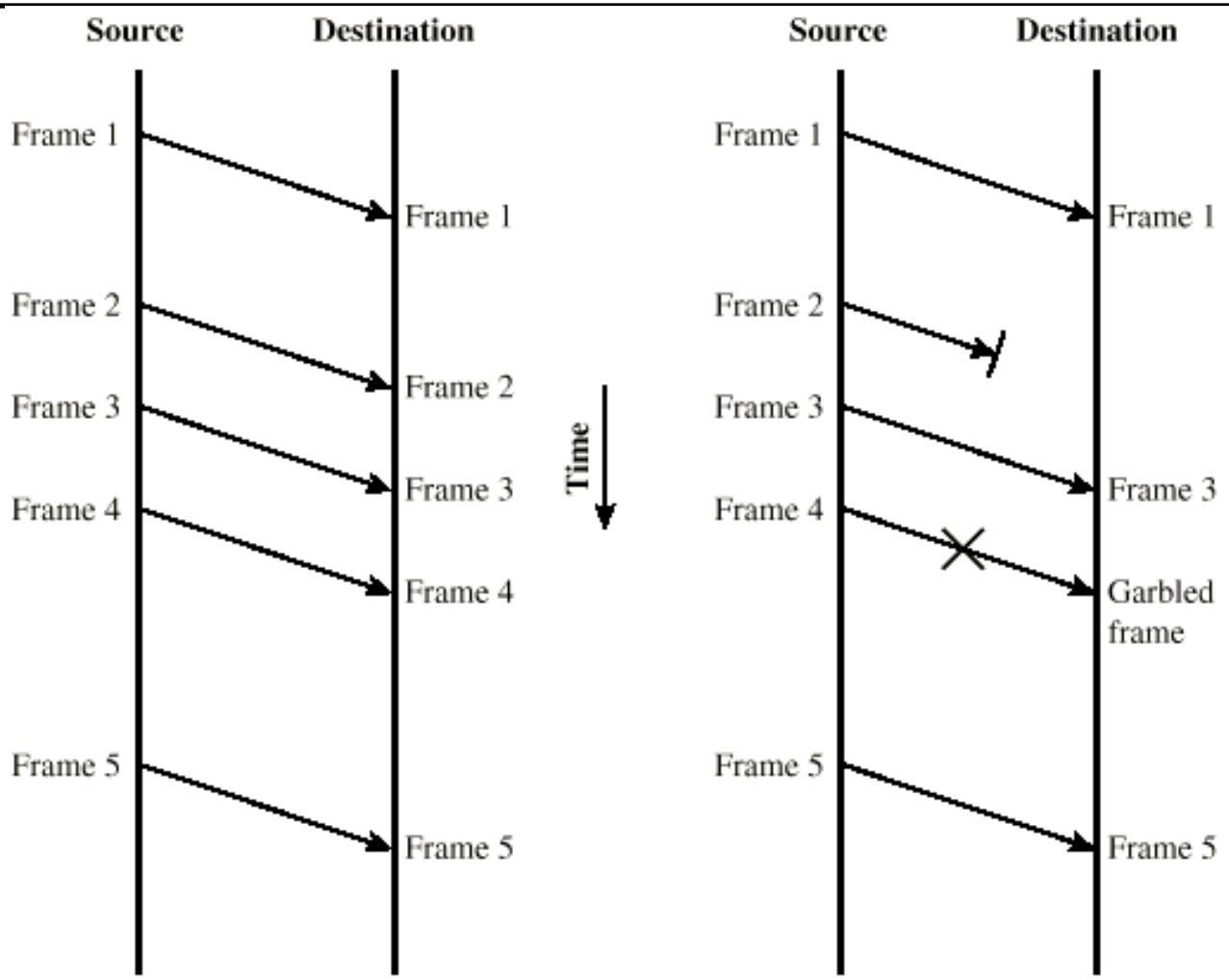
Flow Control

- Ensuring the sending entity does not overwhelm the receiving entity
  - Preventing buffer overflow
- Transmission time
  - Time taken to emit all bits into medium
- Propagation time
  - Time for a bit to traverse the link

Flow control *refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.*

Error control *in the data link layer is based on automatic repeat request, which is the retransmission of data.*

Model of Frame Transmission



(a) Error-free transmission

(b) Transmission with losses and errors

Flow & Error Control Mechanisms

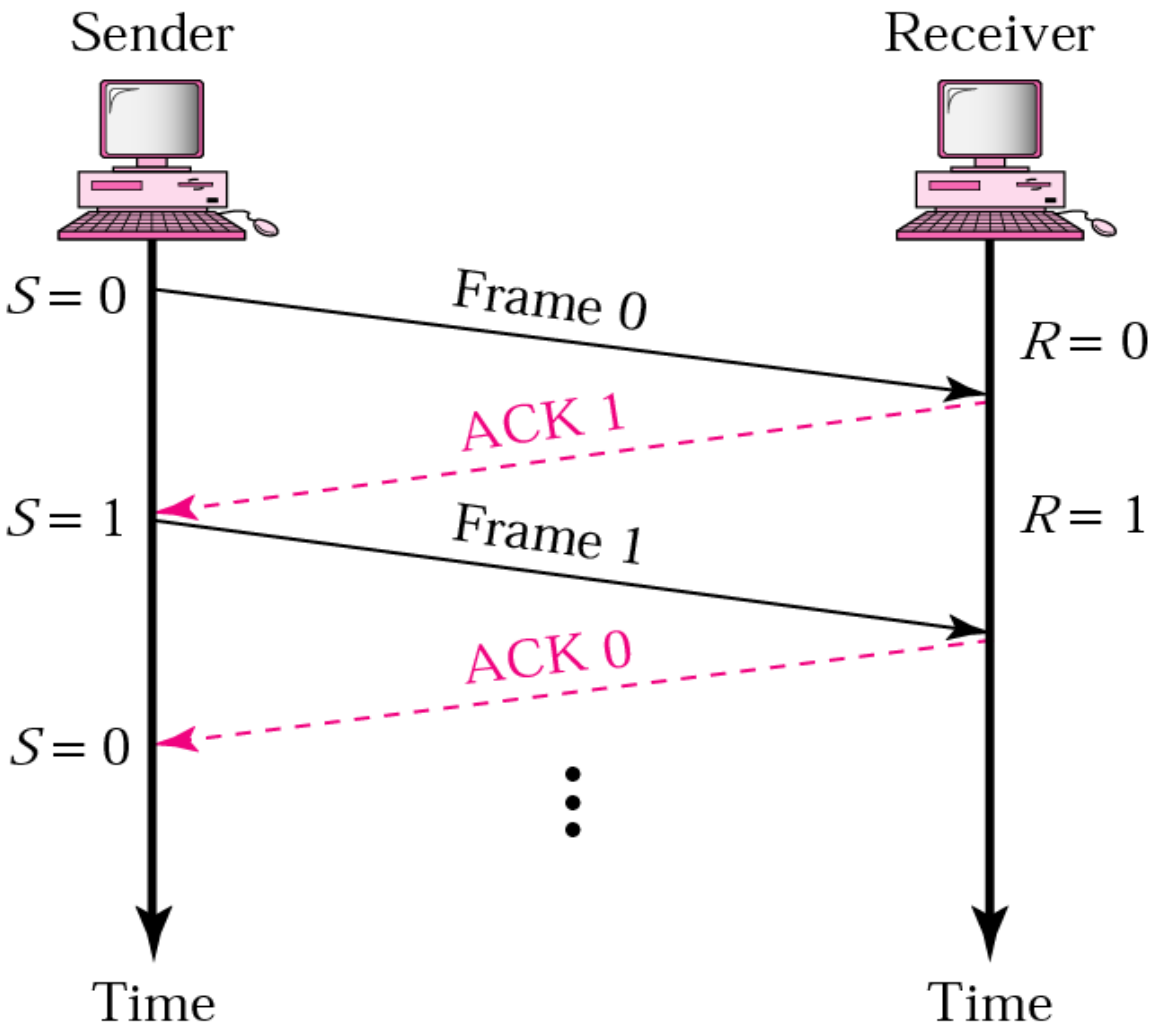
Three Automatic Repeat Request (ARQ) Mechanisms:

- **Stop-and-Wait ARQ**
- **Go-Back-N ARQ**
- **Selective Repeat ARQ**

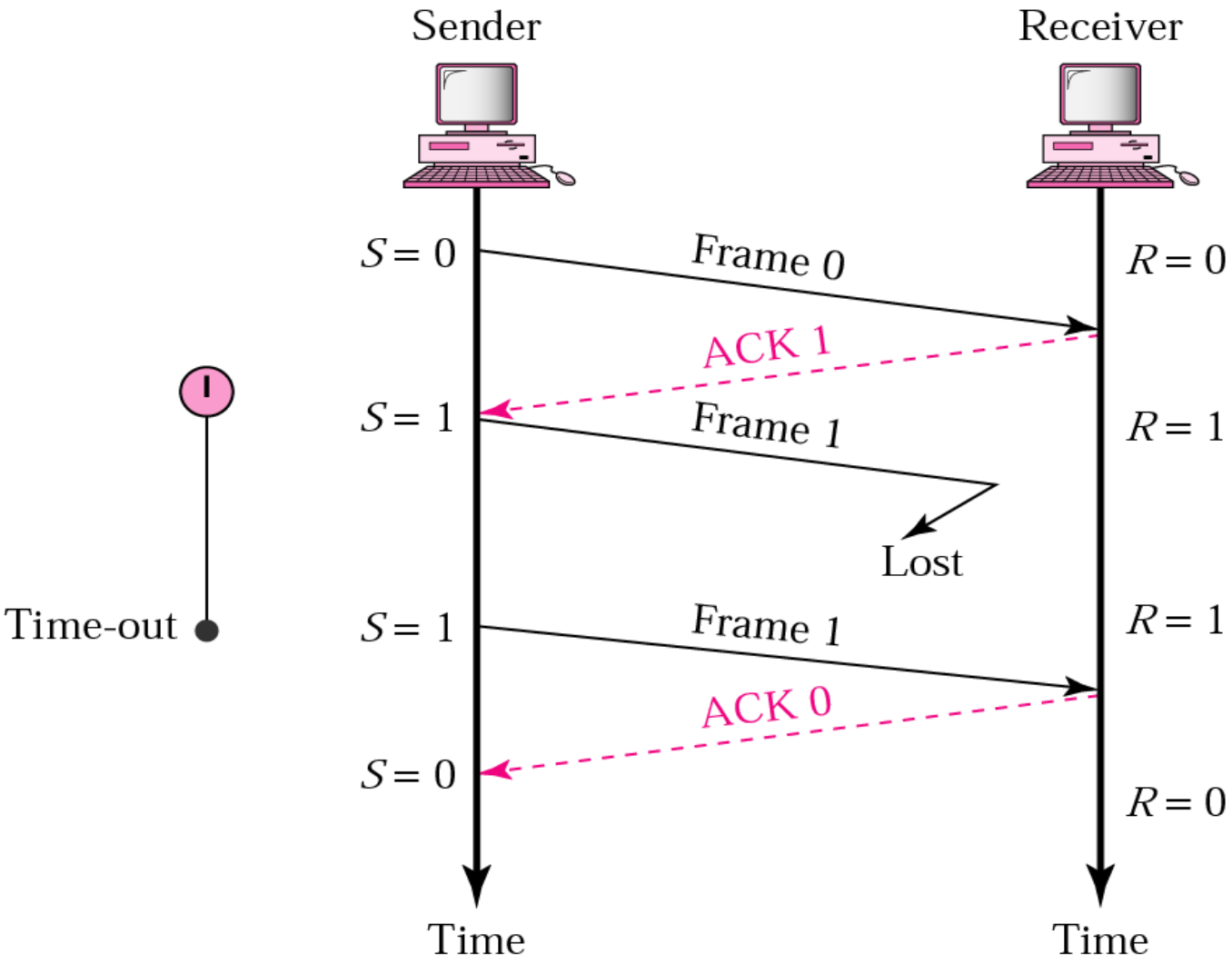
Stop and Wait

- Source transmits frame
- Destination receives frame and replies with acknowledgement
- Source waits for ACK before sending next frame
- Destination can stop flow by not sending ACK
- Works well for a few large frames

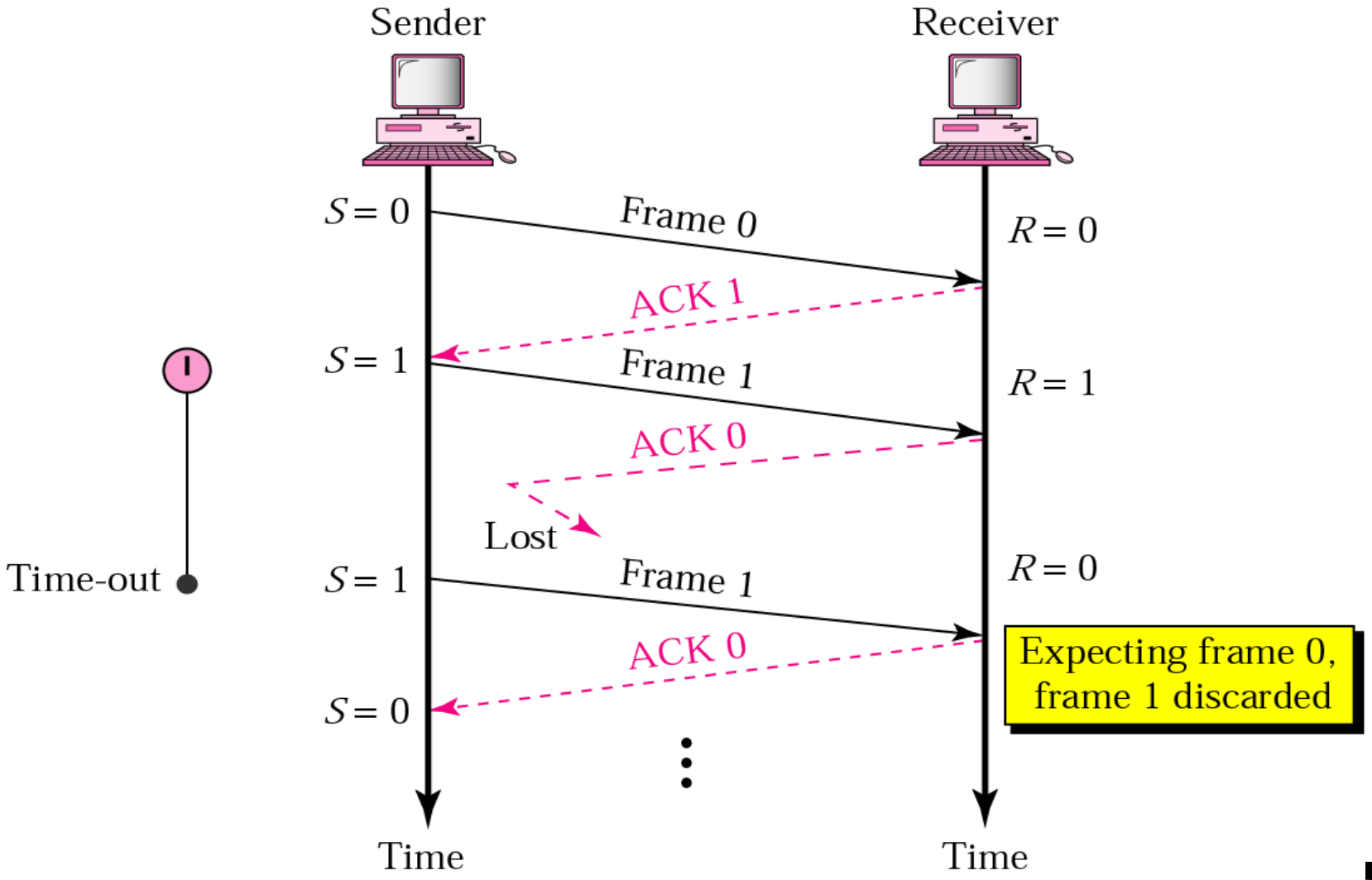
Stop and Wait – Normal Operation



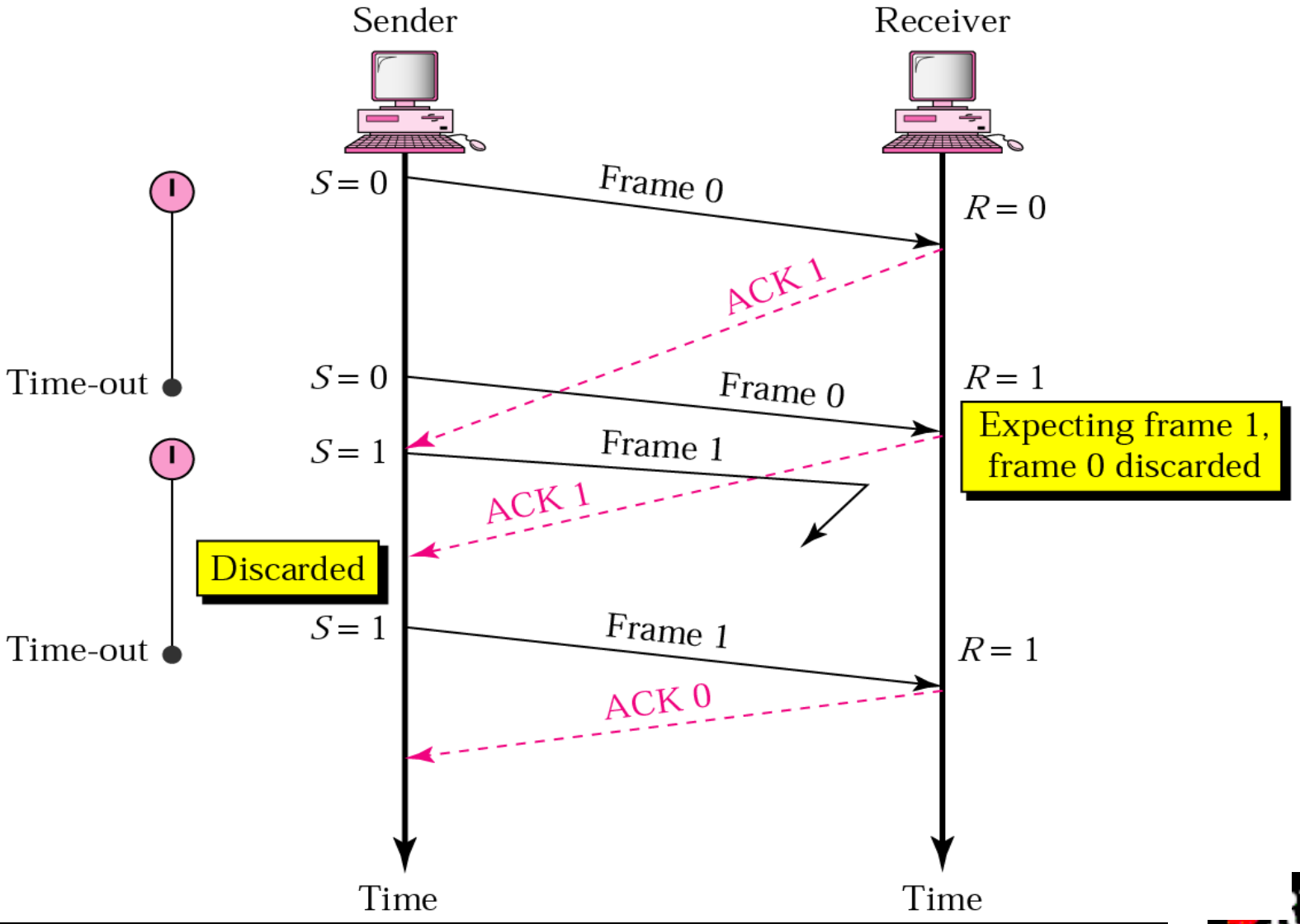
Stop and Wait – Lost or Damaged Frame



Stop and Wait – Lost ACK Frame

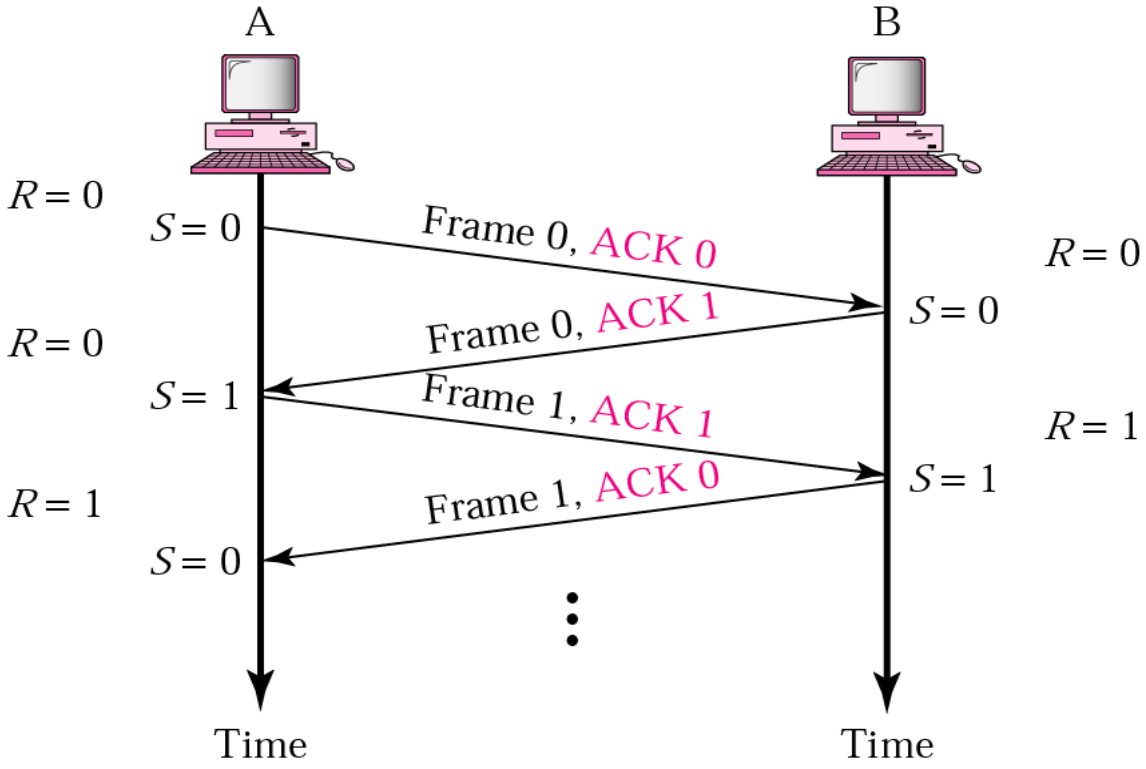


Stop and Wait – Delayed ACK



Piggybacking

- Piggybacking is a method to combine a data frame with an acknowledgment in bidirectional transmission.
- Instead of sending separate data and ACK frames, station A sends a data frame that includes an ACK.
- Can save bandwidth, less overhead.



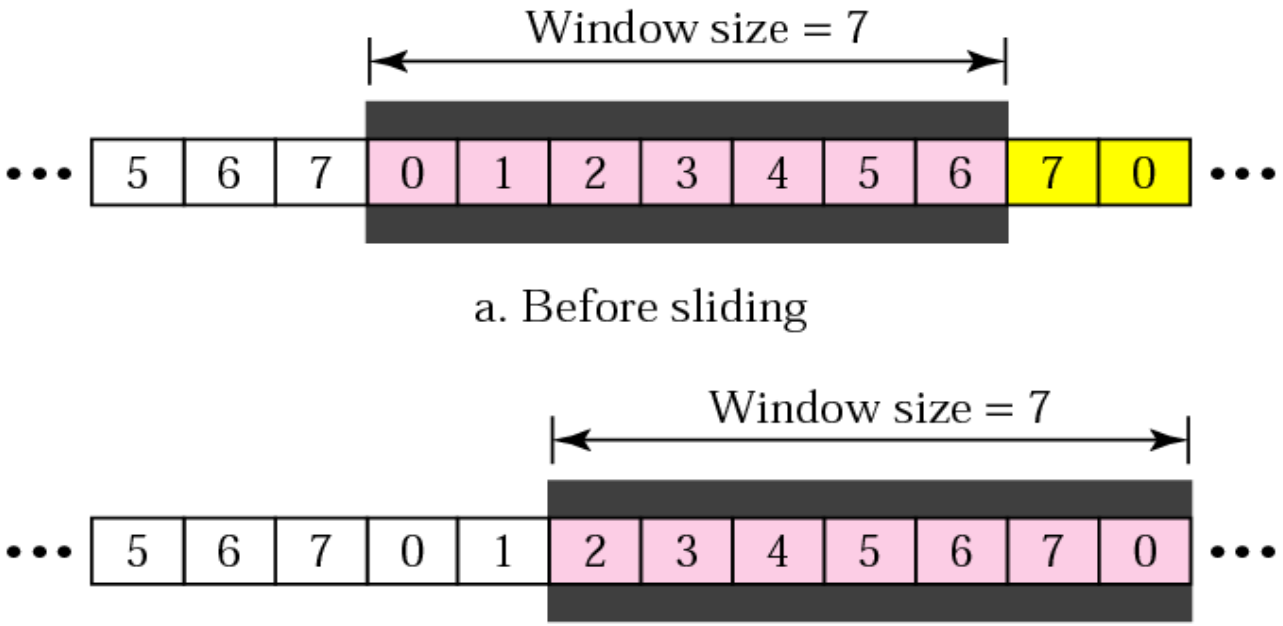
Limitations of Stop-and-wait

- At any point in time for a sender, there is only one frame, that is sent and waiting to be acknowledged.
- This is not a good use of transmission medium
- To improve efficiency, multiple frames should be in transition while waiting for acknowledgement (sol: sliding window protocols)

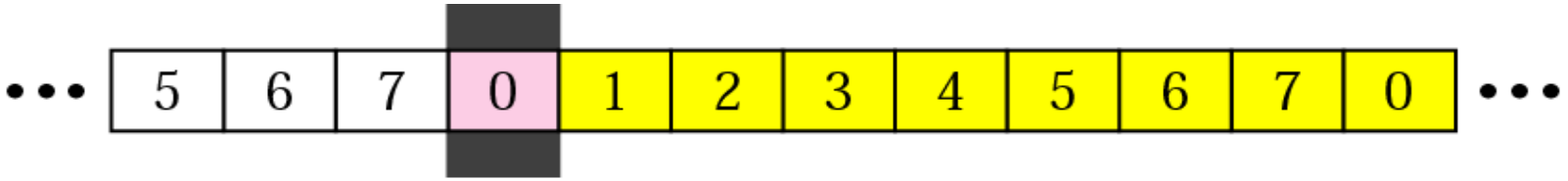
Sliding Windows Flow Control

- Allow multiple frames to be in transit
- Receiver has buffer W long
- Transmitter can send up to W frames without ACK
- Each frame is numbered
- ACK includes number of next frame expected

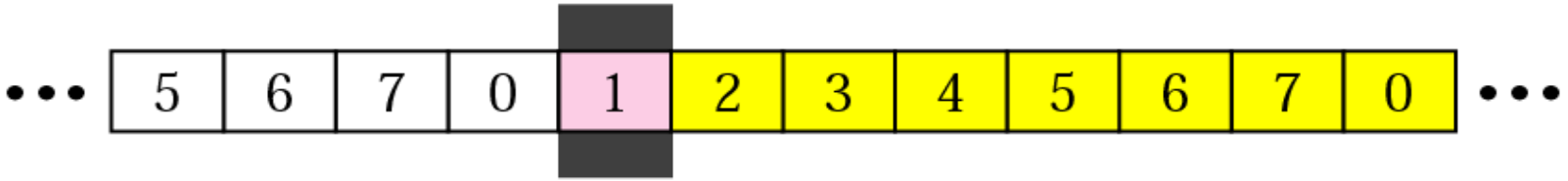
Sender Sliding Window



Receiver Sliding Window



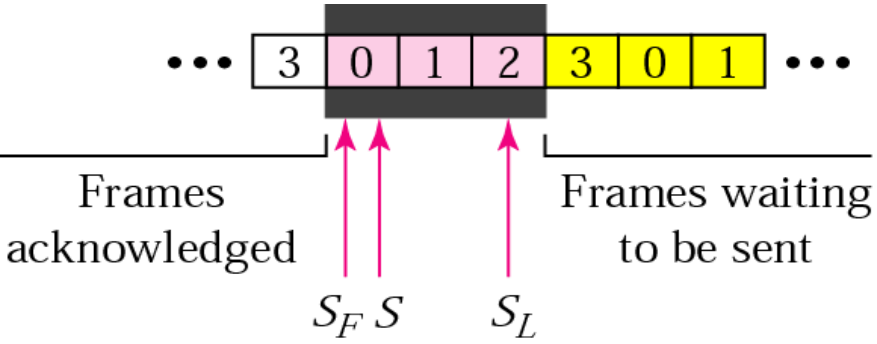
a. Before sliding



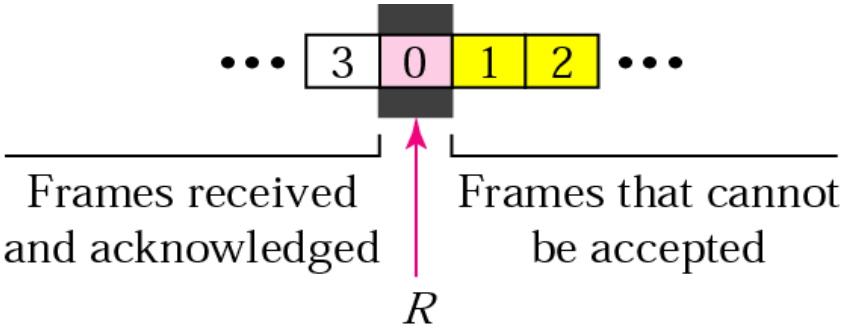
b. After sliding

Sliding Window Variables

S – holds the sequence # of recently sent frame
S_F– sequence # of the first frame in window
S_L – sequence # of the last frame in window
R- sequence # of the frame it expects to receive

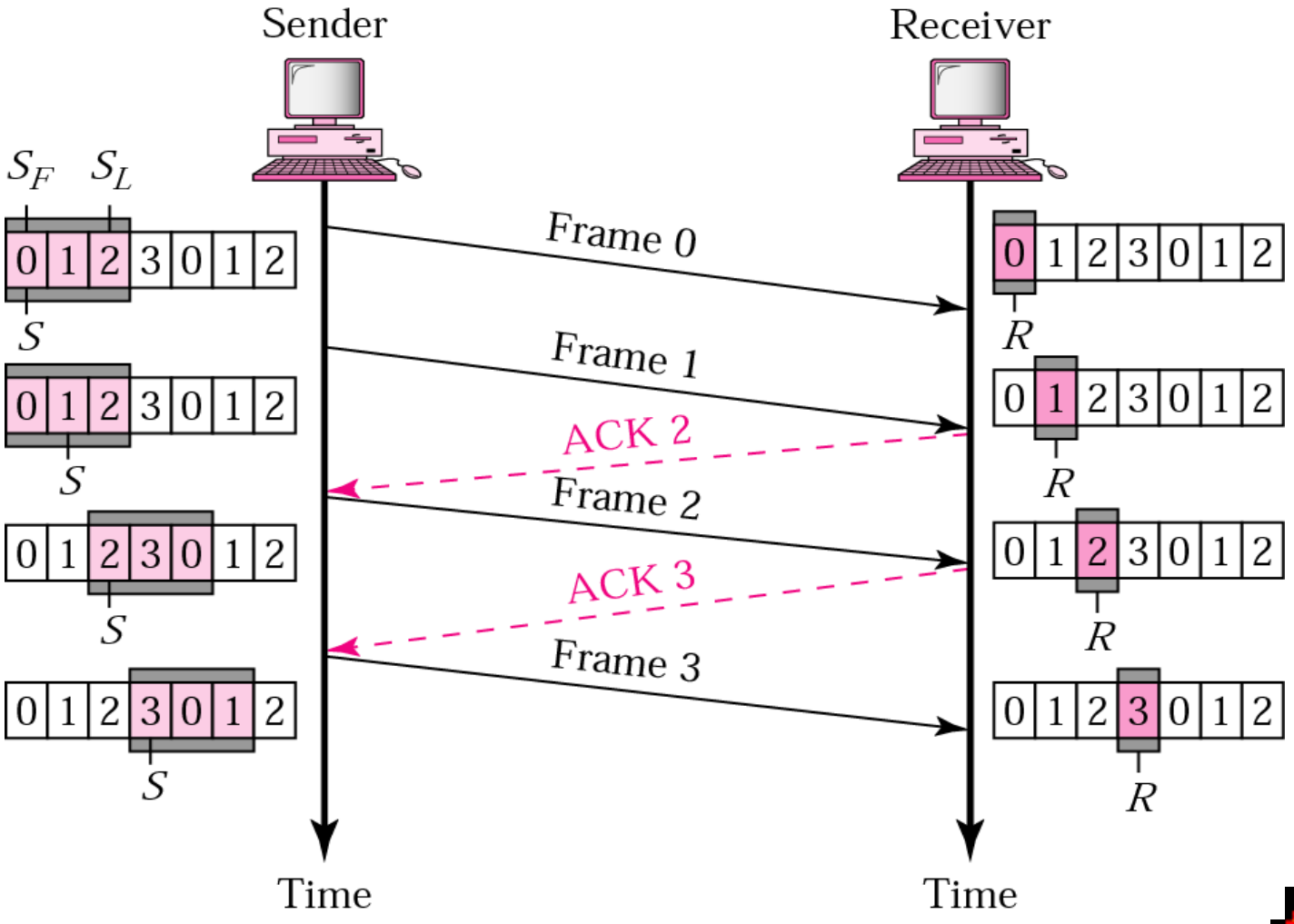


a. Sender window

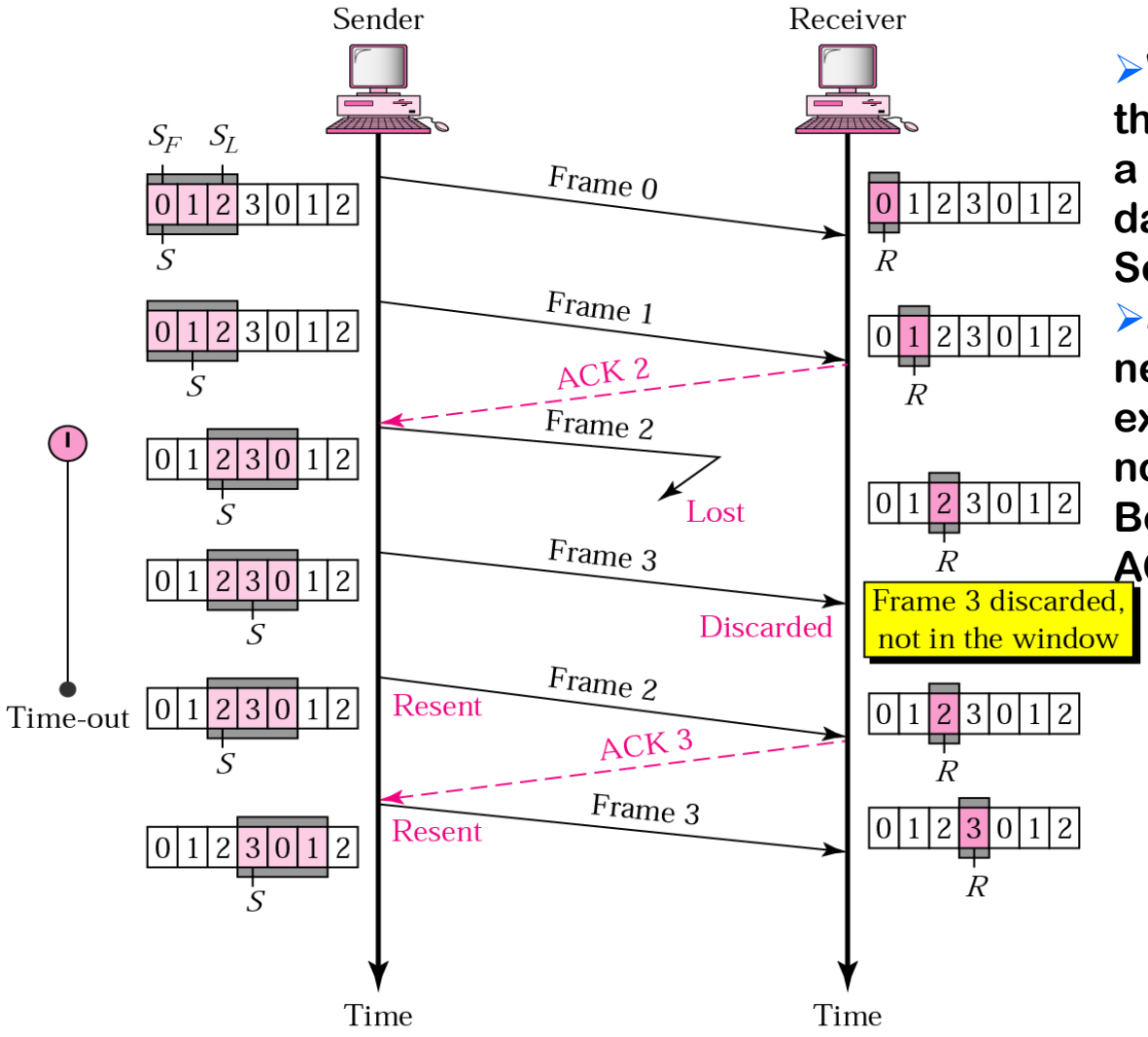


b. Receiver window

Go-Back-N ARQ, normal operation



Go-Back-N ARQ, Lost Frame

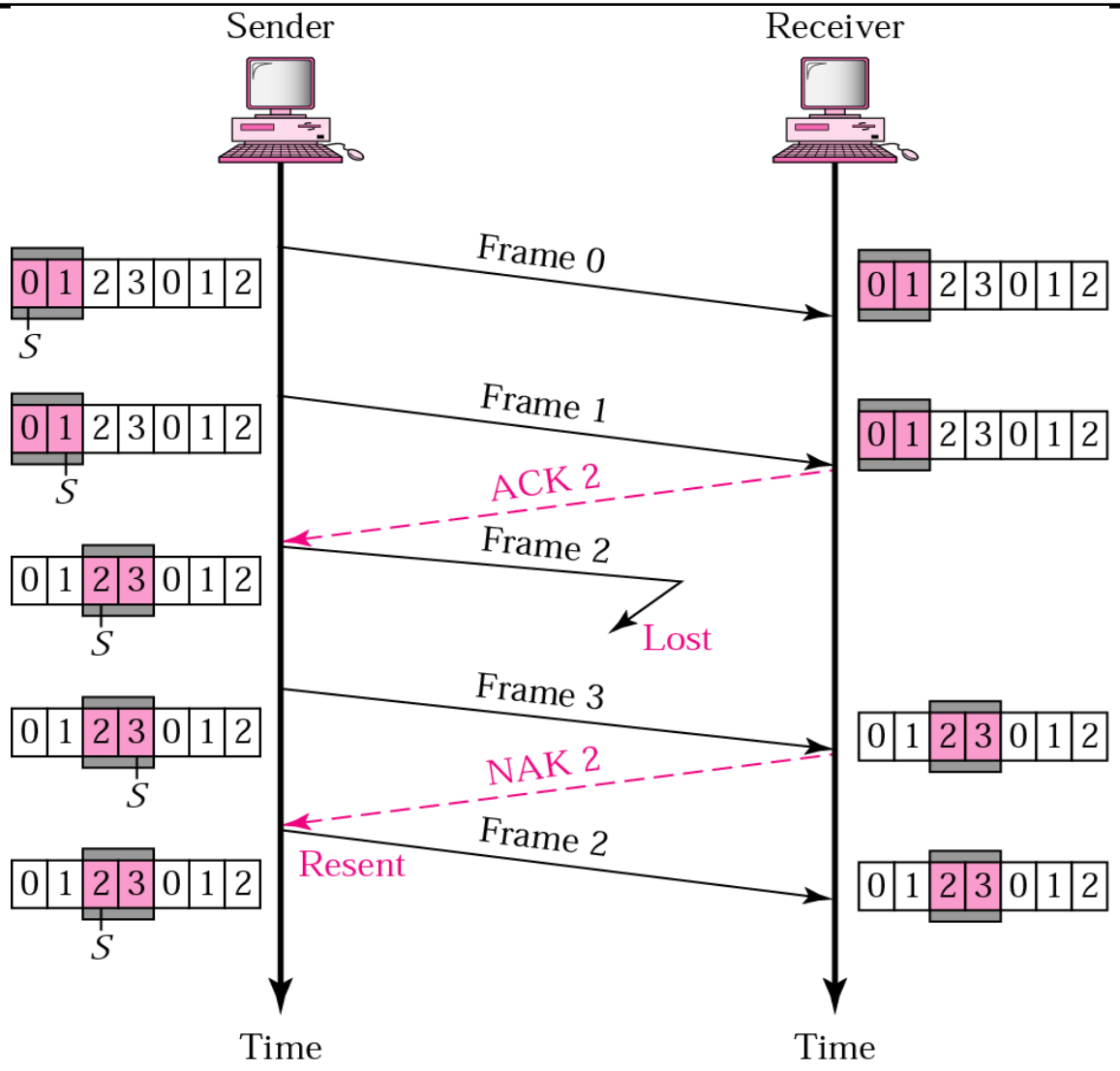


- When a frame is lost/damaged the sender goes back and sends a set of frames starting from the damaged one upto the last one Sent.
- an ACK is damaged/lost – if the next ACK arrives before the expiration of any timer, there is no need for retransmission Because ACKs are cumulative. ACK3 means ACK1 to ACK3.

Selective Reject (Repeat)

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Minimizes retransmission
- Receiver must maintain large enough buffer
- More complex log in transmitter

Selective Repeat ARQ – Lost Frame



Performance - stop and wait

example: 1 Gbps link, 15 ms e-e prop. delay, 1KB packet:

$$T_{frame} = \frac{L \text{ (packet length in bits)}}{R \text{ (transmission rate, bps)}} = \frac{8kb/pkt}{10**9\ b/sec} = 8 \text{ microsec}$$

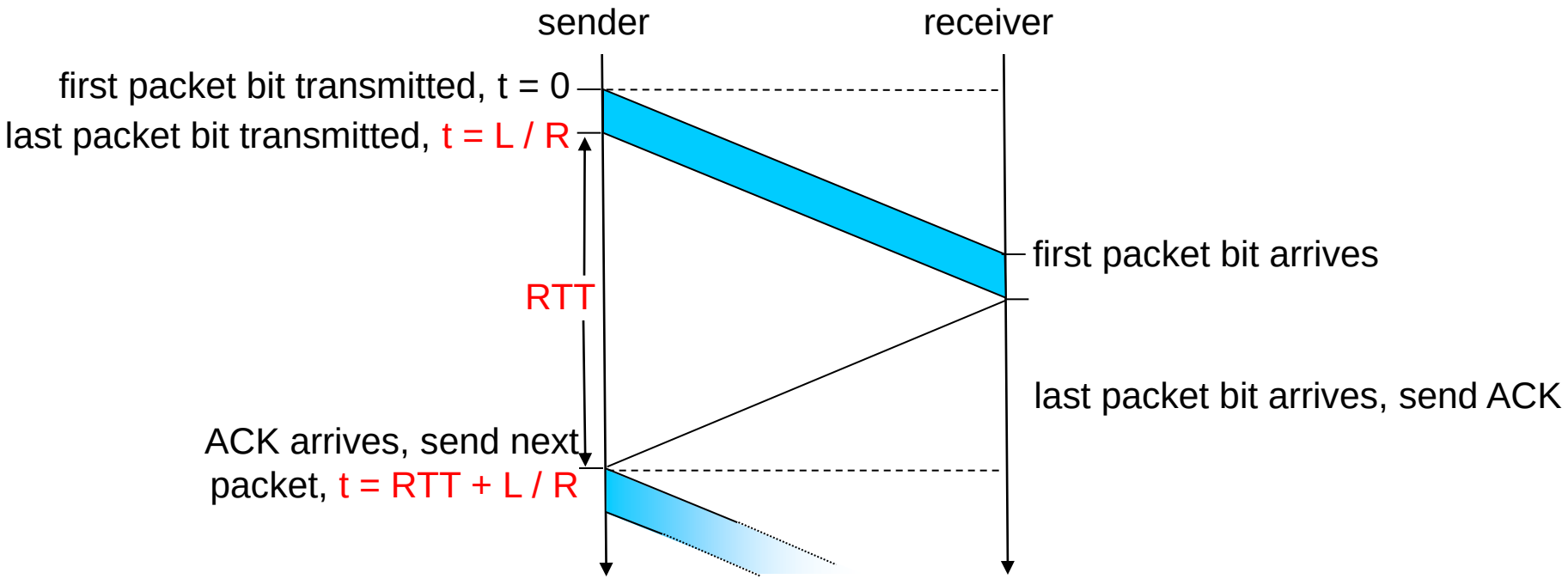
$$U_{sender} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

Or, $$U_{sender} = \frac{1}{2a + 1}$$

T_{Frame} **Transmission time**

- U_{sender}: utilization – fraction of time sender busy sending
- RTT= 2 * (propagation time) (approx)
- a= (propagation time)/ (transmission time)

Performance - stop and wait

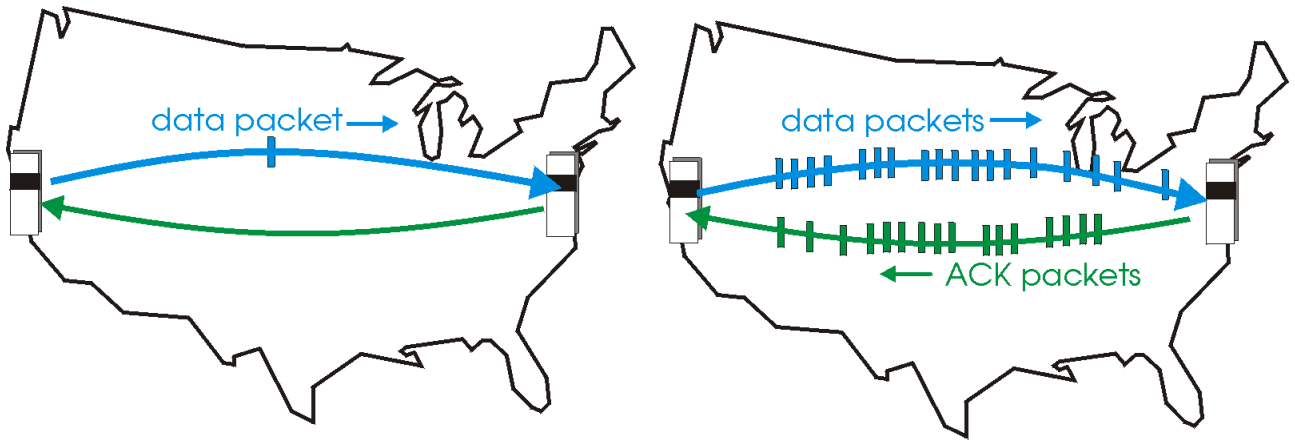


$$U_{sender} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

Pipelined protocols

Pipelining: sender allows multiple, "in-flight", yet-to-be-acknowledged pkts

- range of sequence numbers must be increased
- buffering at sender and/or receiver

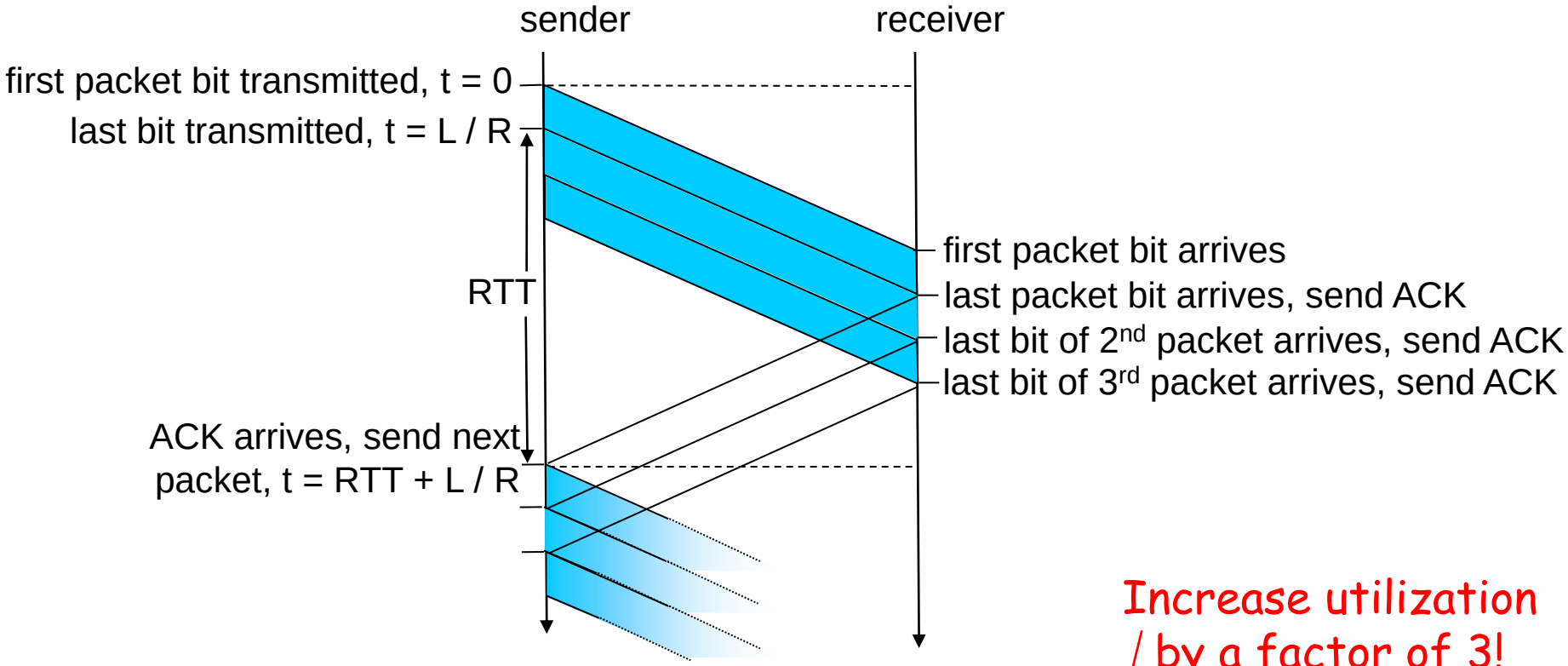


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

- Two generic forms of pipelined (i.e. sliding window) protocols: *go-Back-N, selective repeat*

Pipelining: increased utilization



Increase utilization by a factor of 3!

$$U_{sender} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$

Bandwidth-Delay Product

- **The Bandwidth-delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver.**
- **BDP – an easy an alternative way to find utilization.**
- **BDP= RTT * bandwidth= 2 * (prop time) * bandwidth**

In a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Here, RTT= 20 ms= 20×10^{-3} sec and bandwidth= 1 Mbps = 1×10^{6} bps
The bandwidth-delay product is

$$1 \times 10^{6} \times 20 \times 10^{-3} = 20{,}000 \text{ bits}$$

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5%. For this reason, for a link with high bandwidth or long delay, use of Stop-and-Wait ARQ wastes the capacity of the link.

Summary

- In this lecture, we have understood:
  - Flow control
  - Error control
  - Utilization of Flow control mechanisms
  - Bandwidth-delay product

Next Time

- We will know about
  - Routing
  - Shortest path algorithms

- Suggested Reading:
  - Chapters 12 and 13 (Stallings)